

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS
CURSOS

FORMACIONES

CURSOS

PARA
EMPRESAS

ARTÍCULOS DE TECNOLOGÍA > DATA SCIENCE

Ordenar listas en Python



Alex Felipe

25/09/2020

En mi sistema de registro de productos para un supermercado, recibí una lista con algunos productos:

```
[nombre:chocolate valor:3.45,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,  
nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:arroz valor:8.5]
```

Cada producto es un objeto del tipo Producto, está representado por la siguiente clase:

```
class Producto(object):  
    def __init__(self, nombre, valor):  
        self.__nombre = nombre  
        self.__valor = valor  
  
    def __repr__(self):  
        return "nombre:%s valor:%s" % (self.__nombre, self.__valor)  
  
    def get_nombre(self):  
        return self.__nombre
```

```
def get_valor(self):  
    return self.__valor
```

Note que la lista tiene productos en orden aleatorio. Pero, necesito ordenarlos, es decir, saber cuáles son los productos más caros o los más baratos, o también, ordenarlos por el nombre para facilitar la visualización de los productos. Entonces, ¿cómo lo podemos hacer?

Ordenar lista de números

En Python, cuando necesitamos ordenar listas, podemos utilizar la función [sorted](#), por ejemplo, para una lista de números:

```
numeros = [4,2,6,1,3]  
numeros_ordenados = sorted(numeros)  
  
print numeros  
print numeros_ordenados
```

Tenemos el siguiente resultado:

```
[4, 2, 6, 1, 3]  
[1, 2, 3, 4, 6]
```

Ordenar lista de strings

¡Funciona perfectamente! Pero, ¿también funciona para textos? Veamos:

```
palabras = ["chocolate","galleta", "cafe", "jugo", "porotos", "arroz"]  
palabras_ordenadas = sorted(palabras)  
  
print palabras  
print palabras_ordenadas
```

Entonces, tenemos el siguiente resultado:

```
['chocolate', 'galleta', 'cafe', 'jugo', 'porotos', 'arroz']  
['arroz', 'cafe', 'chocolate', 'galleta', 'jugo', 'porotos']
```

Ordenar lista de objetos

¡Aparentemente funciona con todo! Entonces, también podemos usarlo para objetos:

```
productos = lista_productos()  
productos_ordenados = sorted(productos)  
  
print productos  
print productos_ordenados
```

Veamos el resultado:

```
[nombre:chocolate valor:3.45,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,  
nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:arroz valor:8.5]
```

```
[nombre:chocolate valor:3.45,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,  
nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:arroz valor:8.5]
```

Criterio de ordenación de objetos

Oye, ¿no debería ordenar? Debería, pero primero necesitamos hacernos las siguientes preguntas:

- ¿Cómo ordenamos un objeto del tipo Producto?
- ¿Por el atributo nombre? ¿O por el valor?

Observe que, en principio no sabemos, pues en determinados momentos, podemos querer ordenar por nombre o por valor. En otras palabras, el `sorted`, en principio, ¡también no sabe cómo ordenar nuestro objeto!

Supongamos que queremos ordenar por nombre, ¿cómo el `sorted` sabría eso? ¡No lo informamos en ningún momento!

Por lo tanto, cuando **queremos ordenar un objeto** del tipo `Producto`, **debemos informar ¡por qué atributo se ordenará!**

Definir criterio de ordenación del objeto

Para informar al `sorted` por qué atributo deseamos que ordene, debemos enviar el parámetro `key` con el valor del atributo de la clase deseada. Intentemos por el atributo `valor`:

```
productos_ordenados = sorted(productos, key = Producto.get_valor)
```

Observe que, en este caso, estamos utilizando el `get_valor` justamente por estar escondiendo el atributo. Veamos ahora el resultado:

```
[nombre:chocolate valor:3.45,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,  
nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:arroz valor:8.5]
```

```
[nombre:galleta valor:2.49,  
nombre:chocolate valor:3.45,  
nombre:cafe valor:3.45,  
nombre:jugo valor:4.3,  
nombre:arroz valor:8.5,  
nombre:porotos valor:10.0]
```

¡Vea que ahora nuestros objetos se ordenaron por el valor! En otras palabras, ahora están ordenados desde los más baratos hacia los más caros :D

¿Intentemos también por el nombre? Alteremos y veamos si resulta:

```
productos_ordenados = sorted(productos, key = Producto.get_nome)
```

Entonces, tenemos el siguiente resultado:

```
[nombre:chocolate valor:3.45,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,  
nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:arroz valor:8.5]
```

```
[nombre:arroz valor:8.5,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,  
nombre:chocolate valor:3.45,  
nombre:porotos valor:10.0,  
nombre:jugo valor:4.3]
```

Realizar ordenación reversa

¡Maravilla! ¡Funcionó como lo esperado! Sin embargo, ¿si quisiera el orden inverso? Es decir, desde el más caro hacia el más barato, o entonces, desde la letra "z" hacia la "a", ¿cómo lo haremos?

¡Sencillo! basta agregar el parámetro `reverse` con el valor `True`. Veamos el resultado ordenando por el valor:

```
productos_ordenados = sorted(productos, key = Producto.get_valor, reverse=True)
```



Tenemos el siguiente resultado:

```
[nombre:chocolate valor:3.45,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,
```

```
nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:arroz valor:8.5]
```

```
[nombre:porotos valor:10.0,  
nombre:arroz valor:8.5,  
nombre:jugo valor:4.3,  
nombre:chocolate valor:3.45,  
nombre:cafe valor:3.45,  
nombre:galleta valor:2.49]
```

Ahora con el nombre:

```
productos_ordenados = sorted(productos, key = Producto.get_nombre, reverse=True)
```



Probando el código nuevamente:

```
[nombre:chocolate valor:3.45,  
nombre:galleta valor:2.49,  
nombre:cafe valor:3.45,  
nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:arroz valor:8.5]
```

```
[nombre:jugo valor:4.3,  
nombre:porotos valor:10.0,  
nombre:chocolate valor:3.45,  
nombre:cafe valor:3.45,  
nombre:galleta valor:2.49,  
nombre:arroz valor:8.5]
```

¡Funciona perfectamente! Un recurso muy fácil que no exige innúmeras implementaciones para que funcione :)

Vimos en esta publicación que, cuando queremos ordenar listas en Python, podemos usar la función `sorted`. También vimos que funciona perfectamente para strings o números, sin

embargo, cuando queremos ordenar objetos, debemos informar cómo debe ordenar el objeto, es decir, enviando un atributo del objeto por medio del parámetro key.

¿Qué tal aprender más sobre **Python** y sus diversos recursos? Entonces, ¡mira nuestros cursos de **Python para Data Science** aquí en [Alura](#)!

ARTÍCULOS DE TECNOLOGÍA > DATA SCIENCE

**En Alura encontrarás variados cursos sobre Data Science.
¡Comienza ahora!**

SEMESTRAL

US\$49,90

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

ANUAL

US\$79,90

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

Acceso a todos
los cursos

Estudia las 24 horas,
dónde y cuándo quieras

Nuevos cursos
cada semana

NAVEGACIÓN

PLANES

INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A

CNPJ 05.555.382/0001-33

SÍGUENOS EN NUESTRAS REDES SOCIALES



ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

CURSOS

Cursos de Programación

Lógica de Programación | Java

Cursos de Front End

HTML y CSS | JavaScript | React

Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

Cursos de DevOps

Docker | Linux

Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics |
Liderazgo y Gestión de Equipos | Startups y Emprendimiento